A Method of Obtaining ASIC Schematic Using Scan Chain

Xiaoming Chen [1, a], Yang Liu [1], Songsong Li [2, b]

[1] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, 116024, China

[2] College of Information Engineering, Dalian Ocean University, Dalian, 116023, China

[a]email: chen_xm@dlut.edu.cn, [b]email: lisongsong@dlou.edu.cn

Keywords: Trojan detection; Scan chain; Truth table; Schematic

Abstract. Hardware Trojan is easy to be embedded during IC's design and manufacturing. Mostly used Trojan detection technique based on side-channel approach need a Trojan-free instance, and could not ensure Trojan-free instance's consistent of schematic. With the help of scan chain, one of the widely used design for testability (DFT) technology, a novel approach is proposed to obtain the schematic of Application Specific Integrated Circuit (ASIC). It is verified and tested by formal verification and hardware simulation. This approach not only could help to obtain the Trojan-free instance which has the same schematic as the ordinary design, but also provide a new method to other Hardware Trojan detection techniques.

Introduction

The integrated circuits (IC) are vulnerable to malicious modification. For economic and time to market concerns, modern IC industry cannot avoid involving other vendors' intellectual property(IP), electronic design automation (EDA)software tools, library, or manufacturing in other foundry. This business mode make ICs easily to be embedded Hardware Trojan.

Early in 2003, the America's whitepaper about global migration of the U.S. semiconductor industry has clearly pointed that the untrusted foundry can affect the reliability of ICs and further affect nation's security[1]. Recent years, some side effect news has aggravated people's worried. In the Gulf War, the U.S. army destroy Iraq's radar by the hidden Backdoor of IC[2]. Some Europe's IC vendors can remotely control the system to steal information or invalidate equipment by embedded Trojan[3]. For this reason, more and more research focus on the study of hardware Trojan detection. The most widely used detection methods can be classified into logic testing-based approaches and side-channel analysis-based approaches[4]. Logic testing-based approaches input test vectors to activate the Trojan and recognize its influence. Original design would be modified to improve the detecting coverage. Salmani et al propose a technique to increase the node's toggle rate by inserting dummy flip-flop, making node's toggle rate more than preset threshold. When threshold set as 10-4, it needs adding 16 dummy flip-flops into s38417 benchmark and 0.8% area overhead[5]. Side-channel analysis-based approaches can detect Trojan by comparing the side-channel parameter of pending test and Trojan-free ICs without triggering Trojan[4]. Side-channel analysis-based approaches has become the most widely used and effective technique[6]. Narasimhan at el use the intrinsic relationship between dynamic current and maximum operating frequency to reduce process noise's affect, can achieve a detection resolution of 1.12 percent in±20 percent variation[7].

Side-channel analysis-based approaches need a Trojan-free IC. Agrawal et al propose a reverse-engineer method: comparing IC's layout with original design layer by layer, if same, the testing IC will be regarded as Trojan-free IC[8]. Besides reverse-engineer's disadvantages, this method only can ensure the testing IC's consistency of layout rather than schematic, making this method cannot be used to detect the Trojan embedded at synthesis stage. Li Qing-bao et al propose a method to get state transition diagram base on off-line reverse analysis[9]. This method need to observe all D flip-flops' (DFF) value from the output pins, making this method could only be applied to programmable logic device (PLD) rather than ASIC, because PLD could satisfy this feature. By matching a set of gates used to implement some function, Subramanyan et al propose a

method to get high level netlist with components such as register files, counter, adder, which relevant to datapath. This method can only math about 51% function and 93% logic gates[10].

The main purpose of this paper is to propose a novel method getting ASIC schematic using inherent scanchain which is original used at test stage. Firstly using scanchain, the most widely technology used in design for testability technique (DFT), divides sequential logic circuit into combinational logic block bounded by scanchain and IO ports, making it easy to get ASIC's logic connection. Secondly this paper take formal verification technique to ensure the obtained schematic is same as the original design. Finally this paper use hardware simulation platform ensures its practicability in practical application.

Theory

To make it sample, this paper select small scale ASIC without RAM or ROM as the study object, which only has one scanchain with all DFFs in it. The system block diagram is shown as Fig. 1.

Assuming x_0, x_1, ⋯, x_{m-1} represent ASIC's inputs, y_0, y_1, ⋯, y_{n-1} represent ASIC's outputs, d_0, d_1, ⋯, d_{k-1} represent DFF's inputs, q_0, q_1, ⋯, q_{k-1} represent DFF's outputs, SI represent scanchain's inputs signal, SO represent scanchain's outputs signal, SE represent test enable signal, RST_N represent reset signal.

ASIC is divided into combinational logic block bounded by DFFs, inputs and outputs ports by the inherent scanchain. The ASIC's schematic can be obtained by the following steps: first, obtain the truth table of d_0, d_1, ⋯, d_{k-1} and y_0, y_1, ⋯, y_{n-1} about x_0, x_1, ⋯, x_{m-1} and q_0, q_1, ⋯, q_{k-1}; second, get combinational logic block's schematic by simplifying the truth table, connect combinational logic block with scanchain and IO ports forming ASIC schematic; finally verify the obtained schematic as same as the original design with formal verification .

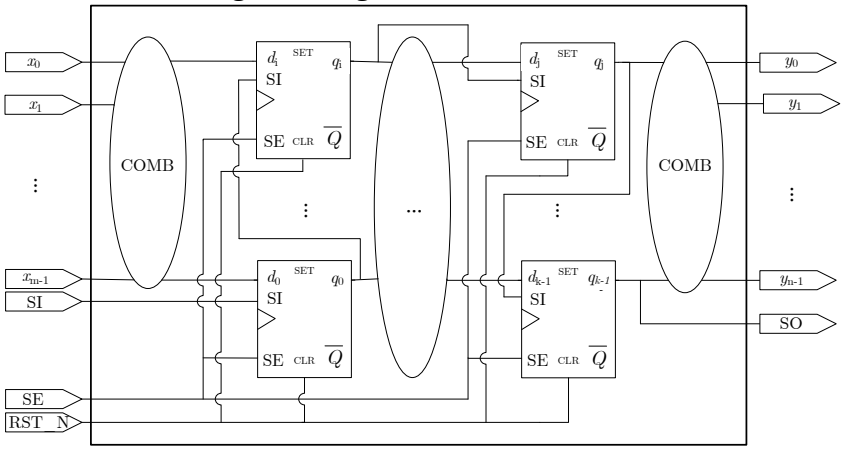

Fig.1 System block diagram with scanchain inserted (omitting clock signal)

Obtaining truth table. Obtaining truth table is implemented by switching between function mode and test mode, obtaining flow is shown as Fig.2(1)Reset ASIC

Reset ASIC by keeping RST_N at low level for about 4 cycles.

(2)In test mode, count DFFS' quantity.

This step is added because the number of DFFs is always unknown in practical application. Input RST_N and SE high level to enable test mode. Input SI high level, wait SO to become high, the DFF's number equals to the waiting cycles.

(3) In test mode, initialize DFF's outputs and ASIC inputs

Keep SE at high level, serially input data from SI, initialize q_0, q_1, ⋯, q_{k-1}. When all DFFs' outputs has been initialized, initialize x_0, x_1, ⋯, x_{m-1}.

(4) In function mode, obtain outputs

Keep SE at low level, After half cycle, obtaining the value of y_0, y_1, ⋯, y_{n-1}.

(5) In test mode, serially output all DFFs' value

Activate SE, enable test mode, Serially output the value of d_0, d_1, ⋯, d_{k-1} from SO.

(6)Exhaust all possible state, obtain truth table

Repeat step (3)-(5), until exhausting all values of x_0, x_1, ⋯, x_{m-1} and q_0, q_1, ⋯, q_{k-1}, obtaining the truth table of y_0, y_1, ⋯, y_{n-1} and d_0, d_1, ⋯, d_{k-1} about x_0, x_1, ⋯, x_{m-1} and q_0, q_1, ⋯, q_{k-1}.

For the AISC of m inputs, n outputs and k DFFS, there are $2^{(n+k)}$ truth tables, and every truth table has as many as $2^{(m+k)}$ miniterm, which need $\left[4+k+(2k+1)\times 2^{m+k}\right]$ cycles to obtain the truth table.

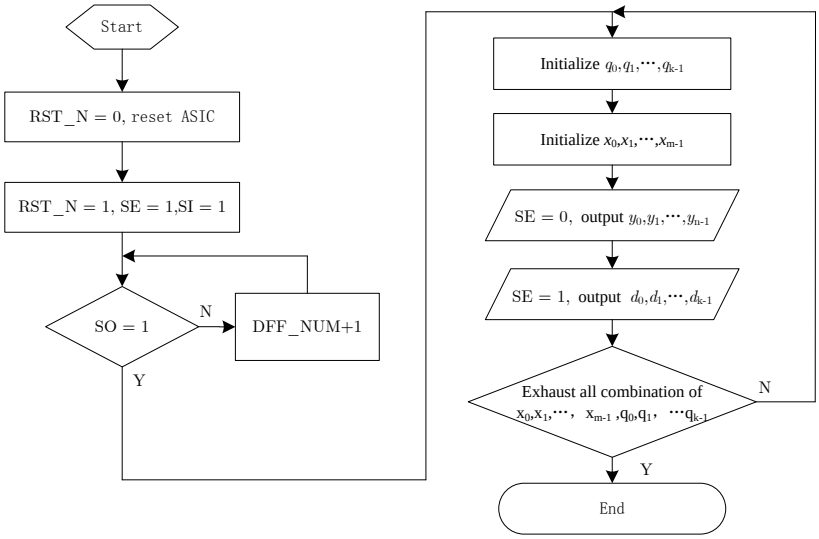

Fig. 2 Obtaining truth table flow

Simplifying truth table. This paper simplifies truth table using Q-M Method proposed by Quine and McCluskey. Compared with Karnaugh Diagram, this method is very suitable for implementation used computer programming. Q-M Simplification method flow is shown as the following steps[11]:

(1) Represent miniterms in binary forms, group miniterms as the number of 1.

(2) The elements from adjacent group can be combined if they have only one different bit. Repeat this steps after regrouping until miniterms cannot be combined any more. All the "and" items which cannot be combined (including the miniterms that cannot be combined), are the prime implicants.

(3) Draw the prime implicants diagram, find the miniterm which only belongs to one prime implicant, this prime implicant which include the miniterm is the necessary prime implicant.

(4) Find the other minimal cover necessary prime implicants by row and column elimination.

Formal verification. There are some differences between the obtained schematic with the original design shown as follows:

(1) The ports relevant to DFT signals such as SI, SO and SE are eliminated, the scan DFFs becomes ordinary DFFs.

(2) The buffer and inverter used to satisfy timing and driving constrains are eliminated.

(3) The logic expression of ASIC's outputs and DFFs' inputs are simplified.

Because of these differences, to ensure the obtained schematic are same as the original design, this paper take formal verification based on assertion, using Mentor Graphics' 0in-formal software, to check their equivalence.

During verification, instantiate the obtained schematic in original design, define following assertions:

(1) The values of obtained ASIC's outputs are same as the original design

(2)The values of obtained DFFs' inputs are same as the original design

(3)The values of obtained DFFs' outputs are same as the original design

The obtained schematic can be thought as same as the original ASIC if the assertion above success in function mode.

Hardware simulation platform

To ensure its practicability in real application, this paper builds a hardware simulation platform shown as Fig. 3. The hardware simulation platform consists of host computer, PCI data acquisition board and field programmable gate array (FPGA).

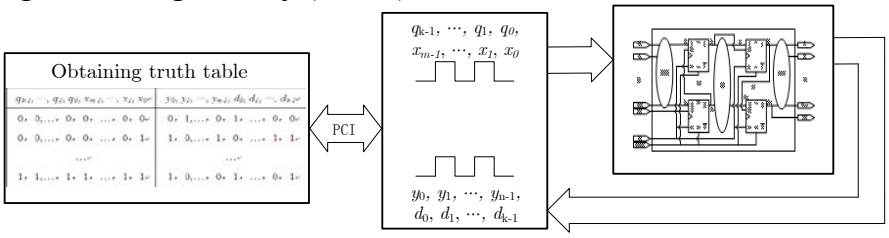

Fig. 3 Hardware system block diagram

FPGA is used to simulate the ASIC of whose schematic is going to be obtained, netlist is loaded in it with scanchain inserted. PCI data acquisition board acts as the interface of host computer and FPGA, sending data to FPGA according to the instruction of host computer, returning FPGA's response to host computer. Host computer is the core part of hardware simulation platform, embedded C program in it.

Results

This paper takes ISCAS89's S27 and a finite state machine (FSM) with 8 states as example. ISCAS89 is benchmark, widely used in many field such as DFT, fault simulation, formal verification, while FSM is a very important abstract form of sequential logic circuits[12], so these two circuits are selected as the study object. S27 and FSM's parameter are shown in Table 1.

Table 1 S27 and FSM's parameters

Circuits	S27	FSM
Inputs numbers	4	1
Outputs numbers	1	1

Obtained DFF's count and truth table are shown as Table 2. To make it sample, truth table are shown in decimal miniterm, when changed to binary form, q_{k-1} represents the most significant bit, x_0 represents the less significant bit.

Table 2 DFF's number and truth table

Circuits		S27	FSM
DFF's number		3	3
DFF's inputs	d_0	1，3，5，7，11，15，17，19，21，23，25，27，29，31，33，35，37，39，43，47，49，51，53，55，57，59，61，63，65，67，69，71，73，75，77，79，81，83，85，87，89，91，93，95，97，99，101，103，105，107，109，111，113，115，117，119，121，123，125，127	1，3，5，7，9，11，13，15
	d_1	……	……
	d_2	……	……
Outputs	y_0	0，1，2，⋯	7，9，11，13，15

Simplify miniterms using Q-M method, obtaining y_0，y_1，⋯，y_{n-1} and d_0，d_1，⋯，d_{k-1}'s logic expression about x_0，x_1，⋯，x_{m-1} and q_0，q_1，⋯，q_{k-1}, shown in Table 3.

Table 3 Logic expression

Circuits		S27	FSM
DFF's inputs	d_0	$x_0q_2 + x_0x_1 + x_0\overline{x_3} + q_0x_0$	x_0
	d_1	$\overline{x_1}x_3\overline{q_0}q_2 + \overline{x_0}q_0q_1$	$\overline{q_1}q_0 + q_1x_0 + q_2x_0$
	d_2	$q_2\overline{x_2} + x_1\overline{x_2}$	$q_1q_0\overline{x_0}$
Outputs	y_0	$\overline{\overline{x_1}x_3\overline{q_0}q_2 + \overline{x_0}q_0q_1}$	$q_1q_0x_0 + q_2x_0$

According to the logic expression of y_0, y_1, …, y_{n-1} and d_0, d_1, …, d_{k-1}, the AISC's schematic can be obtained as Fig 4 and Fig 5.

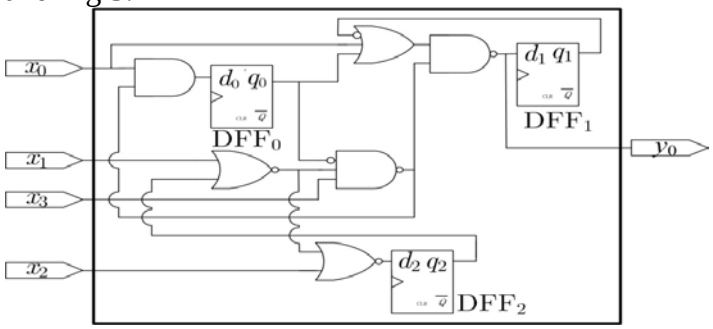

Fig 4 S27's schematic (omitting clock and reset signals)

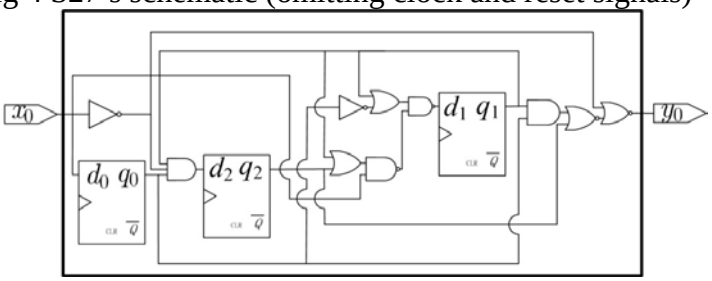

Fig 5 FSM's schematic (omitting clock and reset signals)

It can be found that: when truth table has more variables, it will consume too much time and memory to simplify the truth table, as the original simplifying method has a very low efficiency. This problem can be solved by the improved Q-M Method, such as that XU Jun-pin proposes a method which sets the combination rules to decrease the combination times, structure the bipartite graph which consists of prime implications and minterms, obtain the least prime implicants containing all of the minterns by heuristic approach, which is the minimum coverage of logic functions. This method only needs about 3000 seconds to simplify 19 variables[13].

Formal verification's results are shown as Table 4 and Table 5.

Table 4 s27's Formal verification results

	Defined assertion	**Passed assertion**
ASIC outputs' consistency	1	1
DFF inputs' consistency	3	3
DFF outputs' consistency	3	3

Table 5 FSM's verification results

	Defined assertion	**Passed assertion**
ASIC outputs' consistency	1	1
DFF inputs' consistency	3	3
DFF outputs' consistency	3	3

It can be found that: when truth table has more variables, it will consume too much time and memory to simplify the truth table, as the original simplifying method has a very low efficiency. This problem can be solved by the improved Q-M Method, such as that XU Jun-pin proposes a method which sets the combination rules to decrease the combination times, structure the bipartite graph which consists of prime implications and minterms, obtain the least prime implicants containing all of the minterns by heuristic approach, which is the minimum coverage of logic functions. This method only needs about 3000 seconds to simplify 19 variables[13].

Conclusion

This paper propose an effective method to obtain ASIC schematic using inherent scanchain, which makes complex sequential logic issue become simple combinational logic issue. Experiments on the benchmark s27 and FSM circuit show this method can obtain ASIC's schematic without the disadvantage of reverse-engineer. This method can have two contributions to hardware Trojan detection: a) selecting a Trojan-free ASIC which has the same schematic as original design, b) provide new methods to detect Trojan by comparing pending test ASIC with Trojan-free ASIC's truth table or analyzing the obtained ASIC's schematic. This paper still have some problem to be solved, such as that scanchain inserted is very simple, the truth table simplification has very low efficiency, which need further study.

Acknowledgments

This work was performed at Electronics Science and Technology at Dalian University of Technology, China. We thank the support given by NSFC grant 61340050 and Dalian scientific and technological project 2012F11GH038.

References

[1] Lieberman J I. White paper: National security aspects of the global migration of the us semiconductor industry [R]. Washington: United States Senate Armed Services Committee, 2003.

[2] Jang Ping, Li Dong-jing. Information Countermeasure [M]. Peking: Tsinghua University Press, 2007 8-11.

[3] Adee S. The hunt for the kill switch [J]. IEEE Spectrum, 2008 45(5) 34-39.

[4] Chakraborty R S, Narasimhan S, Bhunia S. Hardware Trojan: Threats and emerging solutions[C]. High Level Design Validation and Test Workshop. San Francisco, 2009 166-171.

[5] Salmani H, Tehranipoor M, Plusquellic J. A novel technique for improving hardware trojan detection and reducing trojan activation time[J]. IEEE Trans. on VLSI Systems, 2012 20(1) 112-125.

[6] Liu Hua-feng, Luo Hong-wei, Wang Li-we. Survey on Hardware Trojan Horse[J]. Microelectronics, 2011 41(5) 709-713.

[7] Narasimhan S, Du D, Chakraborty R, et al. Hardware Trojan detection by multiple-parameter side-channel analysis[J]. IEEE Trans. on Computers, 2013 62(11)2183-2195.

[8] Agrawal D, Baktir S, Karakoyunlu D, et al. Trojan detection using IC fingerprinting[C]. IEEE Symposium on Security and Privacy, 2007. Berkeley, 2007 296-310.

[9] Li Qing-bao, Zhang Ping, Zhao Rong-cai, et al. Data collecting algorithm for reverse analysis of synchronous logic PLD[J]. Computer Engineering, 2008 34(16) 10-12.

[10] Subramanyan P, Tsiskaridze N, Pasricha K, et al. Reverse engineering digital circuits using functional analysis[C].Proceedings of the Conference on Design, Automation and Test in Europe. 2013 1277-1280.

[11] Yan Shi. Fundamentals of digital electronic technology [M]. Peking：Higher education press，200648-51

[12] Chen Zhi-feng, Li Qing-bao, Zeng Guang-yu. Research on sequential PLD security vulnerability detection method[J]. Computer Science, 2012, 39(5) 53-56.

[13] Xu Jun-ping, Cheng Li-xin. Improved Q-M Method for Simplification of Logic Functions[J]. Computer Engineering, 2011, 37(20) 30-32.